Solving ukasiewicz -terms

Citation for published version:
Kalorkoti, K 2018, 'Solving ukasiewicz -terms', *Theoretical Computer Science*, vol. 712, pp. 38-49.
https://doi.org/10.1016/j.tcs.2017.11.002

Digital Object Identifier (DOI):
10.1016/j.tcs.2017.11.002

Solving Łukasiewicz μ-terms

K. Kalorkoti

School of Informatics, University of Edinburgh, 10 Crichton Street
Edinburgh EH8 9LE, U.K. (kk@inf.ed.ac.uk)

Abstract

Łukasiewicz μ-calculus was introduced by Mio and Simpson and is an extension of Łukasiewicz logic, introducing scalar multiplication and least as well as greatest fixed points. A key question is how to evaluate terms of this calculus, i.e., find the values of bound variables occurring in a term. In this paper we provide an algorithm that is single exponential in the size of the term (this takes into account the size of rationals occurring in the term and the interpretation of free variables, the number of operators as well as the number of bound variables). We also show that the solutions are polynomially bounded in the size of the input term and interpretation of free variables. The core technique used is the solution of a set of affine fixed point equations with inequalities as side conditions for which a polynomial time algorithm is given. The techniques introduced here may be of wider interest in model checking and distributive systems.

Keywords: Fixed points, probabilistic μ-calculus, algorithm for evaluating μ-terms.
2010 MSC: 03C80, 68Q60, 68W05, 03D15

1. Introduction

Mio and Simpson [1] introduced the set of Łukasiewicz μ-terms given by

$$x \mid qe \mid e_1 \sqcup e_2 \mid e_1 \sqcap e_2 \mid e_1 \oplus e_2 \mid e_1 \odot e_2 \mid \mu x.e \mid \nu x.e$$

where x denotes a variable and q a rational number from $[0,1]$. We refer to μ, ν as quantifiers and the associated variables as being bound, the rest being free.

The free variables are given values from the rationals by an interpretation ρ. The semantics of such terms are given by:

$$\llbracket x \rrbracket_\rho = \rho(x), \qquad \llbracket qe \rrbracket_\rho = q\llbracket e \rrbracket_\rho$$

$$\llbracket e_1 \sqcup e_2 \rrbracket_\rho = \max\{\llbracket e_1 \rrbracket_\rho, \llbracket e_2 \rrbracket_\rho\}, \qquad \llbracket e_1 \sqcap e_2 \rrbracket_\rho = \min\{\llbracket e_1 \rrbracket_\rho, \llbracket e_2 \rrbracket_\rho\}$$

$$\llbracket e_1 \oplus e_2 \rrbracket_\rho = \min\{1, \llbracket e_1 \rrbracket_\rho + \llbracket e_2 \rrbracket_\rho\}, \qquad \llbracket e_1 \odot e_2 \rrbracket_\rho = \max\{0, \llbracket e_1 \rrbracket_\rho + \llbracket e_2 \rrbracket_\rho - 1\}$$

$$\llbracket \mu x.e \rrbracket_\rho = \mathrm{lfp}(a \mapsto \llbracket e \rrbracket_{\rho(a/x)}), \qquad \llbracket \nu x.e \rrbracket_\rho = \mathrm{gfp}(a \mapsto \llbracket e \rrbracket_{\rho(a/x)}).$$

In the above lfp and gfp denote the least and greatest fixed points, respectively, of the monotonic function $a \mapsto \llbracket e \rrbracket_{\rho(a/x)}$; the existence of the fixed points is guaranteed by the Knaster-Tarski Theorem (see Arnold and Niwiński [2]). In examples we will use stand alone constants q since, e.g., they are shorthand for $q\,\nu x.x$.

By the *solution* to a term we mean the unique values of the bound variables that the term denotes. By a *candidate solution* we mean any assignment to the bound variables (from $[0,1]$) that satisfies the term but ignoring quantifiers, that is each value is a fixed point but not necessarily a greatest or least such. For the sake of simplicity we may assume that the bound variables are given distinct names; so $\mu x.(\nu y.(y \oplus 1/2) \odot x)$ is used rather than $\mu x.(\nu x.(x \oplus 1/2) \odot x)$. In the following we will use σ to denote an unknown quantifier μ or ν. The number of operators of a term e is the number of occurrences of any one of $\sqcup$, $\sqcap$, $\oplus$, $\odot$.

We say that a term is *reduced* if whenever it has a sub-term qe where $q \in \mathbb{Q}$ then e is either a variable or a term that involves at least one operator; this rules out sub-terms of the form $q_1 q_2 \cdots q_r e$ where $r > 1$ which can be replaced by qe where $q = q_1 \times q_2 \times \cdots \times q_r$. We also say that a natural number B is a *magnitude bound* for a rational number u/v (where $u, v \in \mathbb{Z}$) if $|u|, |v| \leq B$, it is said to be *strict* if $|u|, |v| < B$. The number B is a (strict) magnitude bound for a term e and interpretation ρ if it is such for all numbers occurring in e and all numbers assigned by ρ. We will prove the following:

Theorem 1.1. *Assume that B is a strict magnitude bound for a reduced term e and interpretation ρ. Assume also that e has m operators and n bound variables. If $m = 0$ then the value of each variable can be represented in $O(n \lg B)$ bits* *and the values of the variables can be found in time* $O(n^2 \lg^2 B)$*. If* $m > 0$ *then the value of each variable can be represented in* $O(m(m+n)^2 \lg B)$ *bits and the values of the variables can be found in time* $2^{O((m+n)(m+\lg(m+n)+\lg\lg B))}$.

Mio and Simpson [1] provide an elegant algorithm for solving terms but their runtime upper bound has non-elementary growth. It is possible that this algorithm has much better runtime. They also point out that the problem can be solved in triple exponential time using the quantifier elimination methods of Ferrante and Rackoff [3] or double exponential time using the decision procedure for linear arithmetic of Boigelot *et al.* [4]. Thus while the bound in Theorem 1.1 is rather high it does represent a significant improvement. This result and its proof is the private communication from the author referred to on p.343 of [1]; the author is grateful to Alex Simpson for raising the question with him. Mio and Simpson [1] also ask if finding the values of the variables can be shown to be in NP$\cap$co-NP in analogy with the modal μ-calculus introduced by Kozen [5]. Resolving this question seems very hard but at least the bound on the size of the variables shows that the solutions are polynomially bounded in the size of the input term e and interpretation ρ.

2. Terms as equation sequences

Let e be a term with an interpretation ρ. We can convert the term to a sequence of equations by the following standard approach. If e is a term without quantifiers the corresponding equation sequence is just $\mu z = e$ where z is a new variable (the quantifier is irrelevant and ν could be used). If $e = qe_1$, where e_1 has a quantifier, let S be the sequence of equations corresponding to e_1 and z_1 the variable on the left hand of the first equation of S. The sequence of equations for e is $\mu z = qz_1, S$ where z is a new variable. Suppose now that $e = e_1 \circ e_2$ where $\circ$ is one of $\sqcup$, $\sqcap$, $\oplus$, $\odot$ and at least one of e_1, e_2 has a quantifier. If e_1 does not have a quantifier then let S be the sequence of equations corresponding to e_2 and z_1 the variable on the left hand of the first equation of S. The equation sequence is $\mu z = e_1 \circ z_1, S$ where z is a new variable. Similarly if e_2 does not have a quantifier. If both e_1 and e_2 have quantifiers let S_1, S_2 be the equation sequences corresponding to e_1, e_2 respectively. Let z_1, z_2 be the variables on the left hand of the first equation of S_1, S_2 respectively and let z be a new variable. Then the equation sequence for e is $\mu z = z_1 \circ z_2, S_1, S_2$ (again the quantifier is not relevant). Finally, if $e = \sigma x.e_1$ and e_1 is free of quantifiers then the sequence is just $\sigma x = e_1$. Otherwise let S be the sequence of equations corresponding to e_1 and z the variable on the left hand of the first equation of S. The sequence of equations for e is $\sigma x = z, S$.

In practice we can avoid many of the extra variables by substituting their values directly. For example the term $e = \sigma_1 x.(x \sqcap \sigma_2 y.(x \sqcup y))$ can be translated as

$$\sigma_1 x = x \sqcap y,$$

$$\sigma_2 y = x \sqcup y.$$

Thus for any term e we obtain a sequence of n equations in n unknowns (n is generally bigger than the number of bound variables in e).

$$\begin{aligned} \sigma_1\, x_1 &= f_1(x_1, x_2, \ldots, x_n),\\ \sigma_2\, x_2 &= f_2(x_1, x_2, \ldots, x_n),\\ &\vdots\\ \sigma_n\, x_n &= f_n(x_1, x_2, \ldots, x_n). \end{aligned} \tag{$\dagger$}$$

Here each f_i is a monotonic function $[0,1] \to [0,1]$ involving the operators $\sqcup$, $\sqcap$, $\oplus$, $\odot$ (which translate to max, min possibly with arithmetic operators) and scalar multiplication. If we denote this system by E and let $r \in \mathbb{Q}$ the notation $E_{[x_1/r]}$ denotes the system obtained by removing the first equation and substituting r for x_1 in the remaining equations.

Lemma 2.1. *Suppose e is reduced and has m operators and b bound variables. Let n be the number of equations in its translation ($\dagger$). Then $n \leq 1 + 2(m+b)$. Moreover the equations have a total of m operators in the terms appearing on the right hand side. Finally the rational numbers appearing in the equations are the same as those that appear in e.*

PROOF. The second and third claims are clear from the nature of the translation so we consider the first part only.

We call a multiplication by a rational *essential* if it occurs as qe' where the sub-term e' has at least one bound variable and denote the number of essential multiplications by $\mathrm{em}(e)$. We claim that $\mathrm{em}(e) \leq m + b$. This is established by induction on $m + b$. If e has no bound variables then we are done since $\mathrm{em}(e) = 0$. Suppose now that $e = qe'$. Since e is reduced, we must have $e' = \sigma x.e_1$ or $e' = e_1 \circ e_2$ where $\circ$ is one of $\sqcup$, $\sqcap$, $\oplus$, $\odot$. If $e' = \sigma x.e_1$ then the claim follows since $\mathrm{em}(e) = \mathrm{em}(e_1) + 1$. Suppose now that $e' = e_1 \circ e_2$. Let m_i and b_i be the number of operators and bound variables respectively in e_i, for $i = 1, 2$. Now $\mathrm{em}(e) = 1 + \mathrm{em}(e_1) + \mathrm{em}(e_2) \leq 1 + (m_1 + b_1) + (m_2 + b_2) = m + b$. A similar argument applies if $e = \sigma x.e_1$ or $e = e_1 \circ e_2$.

We prove by induction on the size of e that $n \leq 1 + m + b + \mathrm{em}(e)$, the bound on n then follows from the preceding paragraph. If e has no bound variables the claim is immediate. Thus we may assume that e has one of the forms qe', $\sigma x.e'$ or $e_1 \circ e_2$. In each case one of our three parameters drops by 1 while the number of equations needed for the translation of e is just one more than the number of equations needed for the corresponding subexpression(s). $\square$

We proceed to translate the semantics for a term to sequences of equations. If S is a subset of $[0, 1]$ we use μS to denote its infimum and νS to denote its supremum. Given a system (†) its solution is defined as follows:

1. If $n = 1$, set
$$S = \{r \in [0, 1] \mid r = f_1(r)\}.$$
Note that $S \neq \emptyset$ since $f_1 : [0, 1] \to [0, 1]$ is monotonic and $[0, 1]$ is a complete lattice under $\leq$. The solution is $\sigma_1 S$.
2. If $n > 1$ then for each $r \in [0, 1]$ apply the substitution $x_1 \mapsto r$ to the last $n - 1$ equations of the sequence to obtain the system $E_{[x_1/r]}$ consisting of the resulting $n - 1$ equations. By induction, $E_{[x_1/r]}$ has a unique solution $(v_{2r}, \ldots, v_{nr})$. Set
$$S = \{(r, v_{2r}, \ldots, v_{nr}) \mid r = f_1(r, v_{2r}, \ldots, v_{nr})\}.$$

Again $S \neq \emptyset$. The solution is $(r, v_{2r}, \ldots, v_{nr})$ where $r = \sigma_1\{s \mid (s, v_{2s}, \ldots, v_{ns}) \in S\}$.

This definition is a direct translation of the semantics for terms; see Kalorkoti [6] where a similar translation is used for the modal μ-calculus. Note that the process could be generalised by allowing the different variables to take values from different complete lattices but we will not pursue this here other than to observe that the methods described below apply to the general situation quite readily.

Remark: Suppose that $T \subseteq [0, 1]$ includes the first coordinate r of the solution to the system and set

$$R = \{(t, v_{2t}, \ldots, v_{nt}) \mid t \in T \text{ and } t = f_1(t, v_{2t}, \ldots, v_{nt})\}.$$

Then it is clear from above that $r = \sigma_1\{t \mid (t, v_{2t}, \ldots, v_{nt}) \in R\}$. Our strategy will be to find a finite such set T and indeed $|T| \leq 2^m$ where m is the number of operators in the term e.

3. Linear Equations with Side Conditions

Consider a set C of inequalities in $x_1, \ldots, x_n$ together with a sequence E of equations

$$\begin{aligned} \sigma_1\, x_1 &= f_1(x_1, x_2, \ldots, x_n), \\ \sigma_2\, x_2 &= f_2(x_1, x_2, \ldots, x_n), \\ &\ \ \vdots \\ \sigma_n\, x_n &= f_n(x_1, x_2, \ldots, x_n), \end{aligned} \tag{‡}$$

where

$$f_i = a_{i1}x_1 + a_{i2}x_2 + \cdots + a_{in}x_n + b_i,$$

for $1 \leq i \leq n$ (the coefficients and each b_i being from $\mathbb{Q}$). The system E, C is either inconsistent or has a unique solution determined by an obvious adaptation of the semantics given in §2. First we identify the solution, if any, of E:

1. If $n = 1$ let

$$S = \{r \in [0,1] \mid r = f_1(r)\}.$$

   If S is empty then the system is inconsistent. Otherwise the solution is $\sigma_1 S$.

2. If $n > 1$ then for each $r \in [0,1]$ substitute $x_1 \mapsto r$ in the last $n-1$ equations of the sequence to obtain the system $E_{[x_1/r]}$ consisting of the resulting $n-1$ equations. If $E_{[x_1/r]}$ has a solution denote it by $(v_{2r}, \ldots, v_{nr})$ and denote the set of all such solutions obtained as r varies by S'. If S' is empty then the system is inconsistent, otherwise set

$$S = \{(r, v_{2r}, \ldots, v_{nr}) \in S' \mid r = f_1(r, v_{2r}, \ldots, v_{nr})\}.$$

   If S is empty then the system is inconsistent. Otherwise the solution is $(s, v_{2s}, \ldots, v_{ns})$ where $s = \sigma_1\{r \mid (r, v_{2r}, \ldots, v_{nr}) \in S\}$.

Finally, if E has a solution $(s, v_{2s}, \ldots, v_{ns})$ and all inequalities in C are satisfied by substituting $(x_1, x_2 \ldots, x_n) \mapsto (s, v_{2s}, \ldots, v_{ns})$ then $(s, v_{2s}, \ldots, v_{ns})$ is the solution to E, C. Otherwise E, C is inconsistent.

The motivation for such systems is as follows. Suppose that e is a term. Each operator $\sqcup$, $\sqcap$, $\oplus$, $\odot$ of e involves taking the maximum or minimum of two arguments, i.e., deciding an inequality involving affine linear expressions. The term e denotes a unique solution which itself determines which of the two arguments can be picked for each operator. Suppose that we know this choice, then the term gives rise to a consistent system (†) whose solution is precisely that of e. This can be established by an obvious induction on the size of e.

Example. Consider the term

$$e = \mu x.\Big(\nu y.(y \odot \big(x \oplus \frac{1}{2})) \sqcup \frac{1}{2}\Big),$$

which is given as a simple example by Mio and Simpson [1]. It can be seen that the solution is $x = 1$, $y = 1$, e.g., by iteration. We can write the term as follows:

$$\begin{aligned} \mu x &= \max\{y, 1/2\}, \\ \nu y &= \max\{0, y + \min\{1, x + 1/2\} - 1\}. \end{aligned}$$

Thus the choice of inequalities is from:

$$
\begin{aligned}
&y \gtreqless 1/2,\\
&1 \gtreqless x + 1/2,\\
&0 \gtreqless y + \min\{1, x + 1/2\} - 1,
\end{aligned}
$$

with the choice in the second inequality determining $\min\{1, x + 1/2\}$. These simplify to

$$
\begin{aligned}
y &\gtreqless 1/2,\\
x &\gtreqless 1/2,\\
y + \min\{1, x + 1/2\} &\gtreqless 1,
\end{aligned}
$$

Once a choice of inequalities is made we can replace each occurrence of max and min in the equations by an affine linear expression in the variables with rational coefficients and constants. A choice of inequalities can be denoted by a triple such as $(\geq, \geq, \geq)$ which indicates that the chosen inequalities are $y \geq 1/2$, $x \geq 1/2$, $y + \min\{1, x + 1/2\} \geq 1$. These simplify to $y \geq 1/2$, $x \geq 1/2, y \geq 0$. The system becomes

$$
\begin{aligned}
\mu x =& y.\\
\nu y =& y,
\end{aligned}
$$

with $C = \{y \geq 1/2,\ x \geq 1/2,\ y \geq 0\}$. The solution to the equations is $x = 1$, $y = 1$ and C is satisfied.

Another possible system is obtained by the choice $(\leq, \leq, \geq)$ which yields the system

$$
\begin{aligned}
\mu x =& 1/2,\\
\nu y =& y + x - 1/2.
\end{aligned}
$$

with $C = \{y \leq 1/2,\ x \leq 1/2,\ y + x \geq 1/2\}$. The solution to the equations is clearly $x = 1/2$, $y = 1$ but of course C is not satisfied.

Naturally there can be more than one correct translation since in the case where the solution makes the arguments to an operator equal we can take the corresponding inequality either way round. One might hope that if the wrong choice of inequalities is made then the resulting system would have no solution but this is not the case. For example suppose we choose $(\leq, \geq, \leq)$ which yields the system

$$\mu x = 1/2,$$

$$\nu y = 0$$

with conditions $C = \{y \leq 1/2,\, x \geq 1/2,\, y \leq 0\}$. The system is clearly consistent.

4. Full translation

In this section we give a method of organising the choices discussed in §3. Consider a term e as in §2 with the corresponding equation sequence (†). We convert (†) to a system of equations E with side conditions C. The equations in E are of the form $\sigma_i x_i = g_i(x_1, x_2, \ldots, x_n)$ where g_i is affine linear in the variables $x_1, x_2, \ldots, x_n$ and affine multi-linear in a finite set of new t-variables that take values from $\{0, 1\}$. The same holds for the terms in C.

Given two functions a, b and a variable t we define $\mathrm{leq}(a, b, t)$ by

$$\mathrm{leq}(a, b, t) = \begin{cases} a \leq b, & \text{if } t = 1; \\ a \geq b, & \text{if } t = 0. \end{cases}$$

This is a place holder until values are provided for the variables in a, b as well as t (which could occur in a or b) at which time a boolean value for $\mathrm{leq}(a, b, t)$ is obtained. Of course if we have a value for t only then we obtain an inequality in $x_1, x_2, \ldots, x_n$ and possibly other t-variables. An alternative is to make one of the inequalities above strict but this does not give us any advantage with the algorithm presented in this paper.

The translation is applied to each f_i in (†) in turn. First set C to be the empty set. The recursive definition of the translation π is as follows, in each case t is a new variable.

1. $\pi(x) = \rho(x)$, where x is a free variable of the original term, else $\pi(x) = x$.
2. $\pi(qe) = q\pi(e)$.
3. $\pi(e_1 \sqcup e_2) = (1 - t)\pi(e_1) + t\pi(e_2)$; $C := C \cup \{\mathrm{leq}(\pi(e_1), \pi(e_2), t)\}$.

4. $\pi(e_1 \sqcap e_2) = t\pi(e_1) + (1-t)\pi(e_2)$; $C := C \cup \{\, \mathrm{leq}(\pi(e_1), \pi(e_2), t) \,\}$.
5. $\pi(e_1 \oplus e_2) = t + (1-t)(\pi(e_1) + \pi(e_2))$, $C := C \cup \{\, \mathrm{leq}(1, \pi(e_1) + \pi(e_2), t) \,\}$.
6. $\pi(e_1 \odot e_2) = t(\pi(e_1) + \pi(e_2) - 1)$; $C := C \cup \{\, \mathrm{leq}(1, \pi(e_1) + \pi(e_2), t) \,\}$.

If there are m operators $\sqcup$, $\sqcap$, $\oplus$, $\odot$ in the term e then we have m t-variables which we denote by $t_1, t_2, \ldots, t_m$. The cost of producing $\pi(e)$ is just linear in the size of e and the size of numbers assigned by ρ. In order to find the values of the bound variables in e we can now assign each t_i its possible values in turn and solve the resulting system of linear equations with side conditions; so we solve at most 2^m systems. This yields a set of candidate solutions that includes the actual solution.

A possible way to avoid the exponential search inherent in trying out all values of the t-variables is to treat them as unknowns and work with underlying multilinear equations as well as the inequalities symbolically. However this is in itself very costly in general.

For a given assignment of the t-variables, we need to put the translation of a term into the form of (‡), i.e., the translation of each term is a linear affine expression in the variables and the translation of each inequality is of the form $l \leq 0$ where again l is a linear affine expression in the bound variables. This will have the possible effect of increasing the size of the numbers involved. In going from a given term to a system (†) none of the numbers changes so we can focus on a term e that appears on the r.h.s. of (†). This will produce a single linear affine form and a set of inequalities. Let $||e||$ denote the maximum absolute value over the numerators and denominators of all numbers that occur in the linear affine forms produced by the translation $\pi(e)$ and a choice of values for the t-variables.

Lemma 4.1. *Assume that B is a strict magnitude bound for a reduced term e on the r.h.s. of (†) and the interpretation ρ. Then $||e|| < 3^m B^{4m+2}$ where m is the number of operators in e.*

proof. Suppose that e has r multiplications by rationals with terms that are not variables. We use induction on $r+m$ to show that $||e|| < 3^m B^{2(r+m+1)}$. The claimed bound follows since $r \leq m$, which can be seen by a simple induction on m. For the main bound note that if $m = 0$ then $r = 0$ and the claim is obvious as e is either x or qx for a free variable x and rational q. Suppose now that $m > 0$, there are 5 cases to consider. If $e = qe_1$ where e_1 is not a variable then $||e|| < B||e_1|| < B \cdot 3^m B^{2(r-1+m+1)} \leq 3^m B^{2(r+m+1)}$. The inequalities in $\pi(qe_1)$ are those of $\pi(e_1)$ which satisfy the bound by the induction hypothesis.

If $e = e_1 \sqcup e_2$ then $\pi(e_1 \sqcup e_2) = (1-t)\pi(e_1) + t\pi(e_2)$ so this is either $\pi(e_1)$ or $\pi(e_2)$, according as $t = 0$ or $t = 1$. The induction hypothesis shows that, for $i = 1, 2$, the rationals in $\pi(e_i)$ all have magnitude bounded strictly by $3^{m_i-1}B^{2(r_i+m_i+1)}$ where m_i is the number of operators in e_i and r_i is the number of multiplications by rationals in e_i with terms that are not variables. There is only one new inequality involved which is $\mathrm{leq}(\pi(e_1), \pi(e_2), t)$. this can be expressed as $\pi(e_2) - \pi(e_1) \leq 0$ or $\pi(e_1) - \pi(e_2) \leq 0$, according as $t = 0$ or $t = 1$. Let a_i/b_i be a coefficient of some given variable in the translation of e_i, for $i = 1, 2$ (or a constant in each case). The corresponding coefficient or constant in $\mathrm{leq}(\pi(e_1), \pi(e_2), t)$ has absolute value $|a_1/b_1 - a_2/b_2|$. Now by induction $|a_1 b_2| + |a_2 b_1| < 2 \cdot 3^{m_1} B^{2(r_1+m_1+1)} \cdot 3^{m_2} B^{2(r_2+m_2+1)} < 3^m B^{r+m+1}$. The bound for $|b_1 b_2|$ follows more simply.

We now deal with the case $e = e_1 \odot e2$. Here $\pi(e_1 \odot e2) = t(\pi(e_1) + \pi(e_2) - 1)$. The inequality for the coefficients of variables follows as above. Let the constants in $\pi(e_1)$, $\pi(e_2)$ be u_1/v_1, u_2/v_2 respectively where $u_1, u_2, v_1, v_2 \in \mathbb{Z}$. Thus the numerator of the constant of $\pi(e)$ is $(u_1 v_2 + u_2 v_1 - v_1 v_2)/v_1 v_2$. By induction $|u_1 v_2 + u_2 v_1 - v_1 v_2)| \leq 3 \cdot 3^{m_1} B^{2(r_1+m_1+1)} \cdot 3^{m_2} B^{2(r_2+m_2+1)} = 3^m B^{r+m+1}$. The bound for the denominator follows more simply. The bound for the inequalities follows similarly. The remaining two cases are straightforward. $\square$

LEMMA 4.2. *Let e be a term with m operators and no variables with B a magnitude bound on the numbers appearing in e. Then $3^m B^{m+1}$ is a bound on the value of e*

PROOF. Straightforward induction on m. $\square$

4.1. An example

Consider again the term

$$t = \mu x.\Big(\nu y.(y \odot \big(x \oplus \frac{1}{2})\big) \sqcup \frac{1}{2}\Big),$$

from Mio and Simpson [1]. The equations are:

$$\begin{aligned} \mu\ x &= (1 - t_1)\, y + \ t_1/2, \\ \nu\, y &= t_2\, (y + t_3 + (1 - t_3)\, (x + 1/2) - 1)\, . \end{aligned}$$

The inequalities consist of

$$\begin{aligned} &\mathrm{leq}(y, 1/2, t_1). \\ &\mathrm{leq}(1, y + t_3 + (1 - t_3)\, (x + 1/2)\, , t_2), \\ &\mathrm{leq}(1, x + 1/2, t_3), \end{aligned}$$

Consider the choice $t_1 = 0$, $t_2 = 0$, $t_3 = 0$. This yields the equations

$$\begin{aligned} \mu\, x &= y, \\ \nu\, y &= 0, \end{aligned}$$

so that $x = 0$ and $y = 0$. The inequalities are

$$\mathrm{leq}(y, 1/2, 0),\ \mathrm{leq}(1, y + x + 1/2, 0),\ \mathrm{leq}(1, x + 1/2, 0).$$

These assert that

$$1/2 \leq y,\ x + y \leq 1/2,\ x \leq 1/2$$

the first of which is inconsistent with $y = 0$.

The choice $t_1 = 1$, $t_2 = 1$, $t_3 = 1$ yields the equations

$$\begin{aligned} \mu\, x &= 1/2, \\ \nu\, y &= y, \end{aligned}$$

so that the solution is $x = 1/2$ and $y = 1$. The inequalities are

$$\mathrm{leq}(y, 1/2, 1), \mathrm{leq}(1, y + 1, 1), \mathrm{leq}(1, x + 1/2, 1)$$

which assert

$$y \leq 1/2,\ 0 \leq y,\ 1/2 \leq x,$$

once again the first of these is inconsistent with $y = 1$.

The choice $t_1 = 0$, $t_2 = 1$, $t_3 = 0$ yields the equations

$$\begin{aligned}\mu\, x &= y,\\ \nu\, y &= y + x - 1/2.\end{aligned}$$

The second equation reduces to $x - 1/2 = 0$ thus $x = 1/2$ and $y = 1/2$. The inequalities are

$$\mathrm{leq}(y, 1/2, 0), \mathrm{leq}(1, y + x + 1/2, 1), \mathrm{leq}(1, x + 1/2, 0).$$

These assert that

$$1/2 \leq y,\ 1/2 \leq y + x,\ x \leq 1/2,$$

all of which hold.

Proceeding in this way we obtain the set $\{(1/2, 0), (1/2, 1/2), (1, 1)\}$ for the candidate solutions. The solutions correspond to the choices $(1, 0, 0)$, $(0, 1, 0)$ and $(0, 1, 1)$ for (t_1, t_2, t_3) with the solution $(1/2, 0)$ also corresponding to $(1, 0, 1)$ and and the solution $(1/2, 1/2)$ to $(1, 1, 0)$; these duplications are removed if we use a strict inequality for one branch in the definition of leq.

5. Algorithm for solving a linear system

First of all we look at the case when a system (‡) with n bound variables is derived from a term e that has no operators $\sqcup$, $\sqcap$, $\oplus$, $\odot$; the bounds here are simpler than for the general case. Under this assumption there are no inequalities and the equations are of the form

$$\begin{gathered}\sigma_1\, x_1 = q_1 x_{i_1},\\ \vdots\\ \sigma_n\, x_n = q_n x_{i_n},\end{gathered}$$

where $q_1, \ldots, q_n \in [0, 1] \cap \mathbb{Q}$ and $i_j \in \{\, 1, \ldots, n \,\}$, for $1 \leq j \leq n$. If $i_n = n$ then it follows from the semantics of §3 that $x_n = 0$ if $\sigma_n = \mu$. On the other hand if $\sigma_n = \nu$ then $x_n = 1$ if $q_n = 1$ otherwise $x_n = 0$. If $i_n \neq n$ we may substitute

$q_n x_{i_n}$ for x_n in all but the last equation, solve these and then the value of x_n is just $q_n x_{i_n}$. We now give an upper bound for the cost of this process. Let B be a strict magnitude bound for all the rationals $q_1, \ldots, q_n$. A straightforward argument shows that the value of each variable has magnitude strictly less than B^n and the overall cost of the algorithm is $O(n^2 \lg^2 B)$. Thus the value of each variable can be represented in $O(n \lg B)$ bits.

Consider now a system (‡) with n bound variables derived from a term e that has $m > 0$ operators. We look for a solution by initially ignoring the inequalities C and focusing on the equations E. This yields either no solution or exactly one solution at which point we check the inequalities. From the last equation we have

$$(1 - a_{nn})x_n = a_{n1}x_1 + \cdots + a_{n,n-1}x_{n-1} + b_n.$$

If $1 - a_{nn} \neq 0$ then $x_n = (a_{11}x_1 + \cdots + a_{n,n-1,}x_{n-1} + b_n)/(1 - a_{nn})$. Hence as soon as the other variables are assigned values the value of x_n will be fixed by the expression given. Following the semantics of §3 it follows that we may substitute the expression into the remaining equations, eliminating x_n, to obtain a system of $n-1$ equations in $n-1$ variables (this claim can be established by a simple induction on n). Suppose now that $1 - a_{nn} = 0$ then there is a solution only if $a_{n1}x_1 + \cdots + a_{n,n-1}x_{n-1} + b_n = 0$ and no matter what values are given to the other variables the value of x_n will, according to the semantics of §3 be $\sigma_n\{\,0, 1\,\}$. Thus we may delete the final equation, add $a_{n1}x_1 + \cdots + a_{n,n-1}x_{n-1} + b_n = 0$ to the side conditions C and substitute the value of x_n into the first $n-1$ equations.

Continuing in this way we either find that the linear equations are inconsistent or obtain the unique solution. To be precise, as the algorithm progresses we either find the only possible value of the current variable or an expression for it in terms of the variables remaining to be processed.

The process is essentially Gaussian elimination. As is well known, a naive analysis leads to exponential growth for the magnitude bound of the resulting coefficients, see §5.5 of von zur Gathen and Gerhard [7]; the approach of

Bareiss [8] leads to polynomial bounds. In order to facilitate the analysis of the algorithm given above we consider a slight variant. We think of the equations as being given in the form

$$A_{10}x_0 + A_{11}x_1 + \ldots + A_{1n}x_n = 0,$$
$$\vdots$$
$$A_{n0}x_0 + A_{n1}x_1 + \ldots + A_{nn}x_n = 0,$$

where $A_{ij} \in \mathbb{Z}$ for $1 \leq n$ and $0 \leq j \leq n$. The new variable x_0 will be set to 1 at the end. We call it *reserved*, as the algorithm proceeds we might designate other variables as being reserved. We start as before, if $A_{nn} \neq 0$ then we eliminate x_n from the system and set aside $A_{nn}x_n = -(A_{n0}x_0 + A_{n1}x_1 + \ldots + A_{n-1,n-1}x_{n-1})$ as the defining equation for x_n. If, on the other hand, $A_{nn} = 0$ then we designate x_n as another reserved variable and record the determined value of x_n as well as add $A_{n0}x_0 + A_{n1}x_1 + \ldots + A_{n,n-1}x_{n-1} = 0$ to the set of inequalities. At the end, assuming the system of equations is consistent, we substitute the values of the reserved variables determined by the process (and 1 for x_0) so that the remaining variables are determined by their defining equations, it then remains to check the inequalities.

Assuming that $|A_{ij}| \leq D$, for $1 \leq i \leq n$ and $0 \leq j \leq n$, the elimination performed using the method of Bareiss [8] costs $O(n^c \lg^c D)$ for some (moderate) c, provided that $D \geq 2$ (we will ensure this later). Moreover the resulting coefficients have absolute value at most $n^{n/2}D^n$ since each such coefficient is a determinant of a sub-matrix of size at most $n \times n$ of the matrix of the input coefficients (Hadamard's bound now completes the claim). Note that this bound applies also to the coefficients of equations added to the set of side conditions C. If the system of equations E is inconsistent we discover this during the process of elimination so the cost in this case is $O(n^c \lg^c D)$. From now on we assume that the system of equations is consistent.

We can put a bound on the value of each variable as follows. Let $x_0, x_{i_1}, \ldots, x_{i_r}$ be all the reserved variables by the end of the algorithm. Let $x_{j_1}, \ldots, x_{j_s}$ be all the non-reserved variables by the end of the algorithm; of course $r + s = n$. If

$s = 0$ then each variable has value 0 or 1, so assume that $s > 0$. We can rewrite the linear equations as

$$\begin{pmatrix} A_{j_1 j_1} & A_{j_1 j_2} & \dots & A_{j_1 j_s} \\ A_{j_2 j_1} & A_{j_2 j_2} & \dots & A_{j_2 j_s} \\ \vdots & \vdots & \ddots & \vdots \\ A_{j_s j_1} & A_{j_s j_2} & \dots & A_{j_s j_s} \end{pmatrix} \begin{pmatrix} x_{j_1} \\ x_{j_2} \\ \vdots \\ x_{j_s} \end{pmatrix} = \begin{pmatrix} L_1 \\ L_2 \\ \vdots \\ L_s \end{pmatrix}$$

where each L_i is a linear combination of the reserved variables with each variable having the same but negated coefficient that it had in the original linear system. Once we assign the reserved variables the value determined for them the resulting system has a unique solution and so the displayed matrix is non-singular. There are at most $n - 1$ reserved variables each of which has a value from $\{0, 1\}$. Hence, after substituting the values of the reserved variables, we have $|L_i| \leq (n - 1)D$. The determinant of the coefficient matrix for the non-reserved variables is bounded by $n^{n/2}D^n \geq (n - 1)D$, while the adjoint has entries bounded by $(n - 1)^{(n-1)/2}D^{n-1}$. It follows that the value of each x_{j_r} has denominator bounded by $n^{n/2}D^n$ and numerator bounded by $(n-1)\cdot(n-1)D\cdot(n-1)^{(n-1)/2}D^{n-1} \leq n^{(n+3)/2}D^n$. Hence the value of each variable has magnitude bounded by $n^{(n+3)/2}D^n$.

Next we need to check the inequalities C as well as any new conditions added. We assume that each inequality in C has been expressed in fraction free form and D is an upper bound on the size of all coefficients and constants that occur. As seen above, the coefficients of the added equalities are bounded by $n^{n/2}D^n$. At the start C has m linear inequalities and by the end of the algorithm we have added at most n new equations to C. Let U_i/V_i be the value of x_i found in solving E where $U_i, V_i \in \mathbb{Z}$, for $1 \leq i \leq n$. As shown above, $|V_i| \leq n^{n/2}D^n$ and $|U_i| \leq n^{(n+3)/2}D^n$, for $1 \leq i \leq n$. It follows that $n^{(n+3)/2}D^n$ is an upper bound on the absolute value of all the integers involved. Consider testing an equality $A_1x_1 + \cdots + A_nx_n = A_0$, similar considerations apply to inequalities. Set $M = V_1 \cdots V_n$ and $M_i = V_1 \cdots V_{i-1}U_iV_{i+1} \cdots V_n$, for $1 \leq i \leq n$. Thus the equality is equivalent to $A_1M_1 + \cdots + A_nM_n = A_0M$. Each of the products can be computed in time $O(n^4 \lg^2 nD)$ using the school method and a divide and conquer approach (our final analysis does not benefit from using faster integer multiplication algorithms). Thus the multiplications for the single equality cost $O(n^5 \lg^2 nD)$ in total. The cost of the sum and comparison is dominated by the cost of the multiplications. Since there are at most $m+n$ inequalities and equalities in C the cost of checking them is $O((m+n)n^5 \lg^2 nD)$. By taking d large enough in the runtime of Bareiss elimination we deduce that the total runtime of the algorithm presented is $O((m+n)n^d \lg^d nD)$.

It remains to express the cost of the algorithm in terms of the input size of the reduced term e. Assume that e is has m operators, n bound variables and B is a strict magnitude bound for e and the interpretation (hence $B \geq 2$). Let (‡) be the system obtained from e and denote the number of equations by N. Then $N \leq 1 + 2(m+n)$, by Lemma 2.1, and $|C| \leq m$ while each coefficient has strict magnitude bound $3^m B^{4m+2}$, by Lemma 4.1. After clearing fractions the coefficients are bounded by $D = 3^{m(N+1)} B^{(4m+2)(N+1)}$. We may now substitute into the magnitude bound found above for the values of the variables, this yields

$$M = (2m+2n+1)^{m+n+2} 3^{m(2m+2n+2)(2m+2n+1)} B^{(4m+2)(2m+2n+2)(2m+2n+1)}.$$

Taking logarithms we see that each value can be represented with $O(m(m+n)^2 \lg B)$ bits, which is the bound on representation length claimed in Theorem 1.1. For the runtime we obtain a bound of $O(m^d(m+n)^{2d+1} \lg^d (m+n)B)$ which can be expressed as $O((m+n)^c \lg^c B)$ for a large enough c.

6. Finding the solution to Łukasiewicz μ-terms

Suppose now that e is a reduced Łukasiewicz μ-term. We convert it to a system (†) denoted by E and from this find a set S of candidate solutions that we know includes the solution to e. Recall that $|S| \leq 2^m$ where m is the number of operators in the term e. We can now find the solution by the recursive algorithm of Figure 1. The correctness of the algorithm follows from the Remark at the end of §2. The runtime of the algorithm is dominated by the cost of finding S

$T \leftarrow \emptyset$

for $t \in \pi_1 S$ **do**

Find (recursively) the solution $(t, v_{2t}, \ldots, v_{nt})$ of $E_{[x_1/t]}$

if $t = f_1(t, v_{2t}, \ldots, v_{nt})$ **then** $T \leftarrow T \cup \{ (t, v_{2t}, \ldots, v_{nt}) \}$

$r \leftarrow \sigma_1 \pi_1 T$

return $(r, v_{2r}, \ldots, v_{nr})$

Figure 1: Identifying the solution to E form the candidates S.

and the loop with the recursion. Finding S for a reduced term e can be done by the algorithm discussed in §5 and costs $2^{O(\lg(m+n)+\lg\lg B)}$, where B is a strict magnitude bound for e and the interpretation ρ. Before analysing the cost of the recursive algorithm we note that the solutions we are seeking all have magnitude bounded strictly from above by M, where M is defined at the end of §5. Thus we can amend the algorithm so that as soon as any candidate solution (during an invocation of the recursion) would have magnitude greater than this then it is skipped. This does not affect the asymptotic runtime. Suppose that E has N equations. When substituting a value t for a variable, x_1 say, to obtain $E_{[x_1/t]}$ we carry out at most N multiplications; this does not affect the asymptotic runtime. All other numbers in E are unchanged. Thus BM is strictly bigger than the magnitude of any number that occurs in any term constructed during the recursive algorithm. Hence the runtime for finding the candidate solutions at any stage is

$$\begin{aligned} O((m+n)^c \lg^c BM) &= O(m^c (m+n)^{3c} \lg^c B) \\ &= 2^{O(\lg(m+n)+\lg\lg B)}. \end{aligned}$$

We may now analyse the recursive algorithm without further reference to the size of coefficients.

Let $L(m, N)$ denote the runtime of the algorithm given above for a system E of N equations and m operators. Now

$$L(m,N) \leq 2^{O(\lg(m+n)+\lg\lg B)} 2^m L(m_1, N-1) = 2^{O(m+\lg(m+n)+\lg\lg B)} L(m_1, N-1),$$

where $m_1 \leq m$. It follows that

$$L(m, N) = 2^{O(N(m+\lg(m+n)+\lg\lg B))} L(\overline{m}, 0),$$

for some $\overline{m} \leq m$ or

$$L(m, N) = 2^{O(m(m+\lg(m+n)+\lg\lg B))} L(0, \overline{N}),$$

for some $\overline{N} < N$. A simple argument, using the bound of Lemma 4.2, shows that $L(\overline{m}, 0) = O(\overline{m}^2 \lg BM))$ and we have $L(0, \overline{N}) = O(\overline{N}^2 \lg^2 BM)$ from §5. Since $N \leq 1 + 2(m+n)$, by Lemma 2.1, it follows that

$$L(m, N) = 2^{O((m+n)(m+\lg(m+n)+\lg\lg B))},$$

which is the runtime bound stated in Theorem 1.1.

6.1. A heuristic for identifying the solution

The process discussed in §3 leads to a set S of candidate solutions for a system (†) derived from a term e with $|S| \leq 2^m$ where m is the number of operators in e. The set S is known to include the actual solution to (†), the key problem is to identify it. Assume that (†) has n equations and that free variables have been replaced with their values given by the interpretation. Consider the algorithm:

$S_{n+1} \leftarrow S$
for $i \leftarrow n$ **downto** 1 **do**
$\quad S_i \leftarrow \{(s_1, \ldots, s_n) \in S_{i+1} \mid s_i = \sigma_i x_i . f_i(s_1, \ldots, s_{i-1}, x_i, s_{i+1}, \ldots, s_n)\}$

A simple argument shows that the solution to the system is in S_1. Thus if S_1 is a singleton set then its element is the solution to e, while if S_1 is empty (for an arbitrary S) then S does not contain the solution to (†). As a practical point, we can stop the algorithm as soon as a singleton set is obtained provided we know that S contains the solution.

Example. Consider the term $e = \mu x.(x \sqcap \nu y.(x \sqcup y))$ from above, yielding the equations.

$$\mu x = x \sqcap y = \min\{x, y\},$$

$$\nu y = x \sqcup y = \max\{x, y\}.$$

Then $S = \{(0,0),(0,1)\}$. The solution to the term is easily seen to be $(0,1)$, e.g., by iterating from $x = 0$. The algorithm sets $S_2 = \{(0,1)\}$ so we can stop here. Note that we could include $(1,1)$ as a candidate solution (this is not included by the algorithm of §5). In this case $S_3 = \{(0,0),(0,1),(1,1)\}$, $S_2 = \{(0,1),(1,1)\}$, and $S_1 = \{(0,1)\}$.

Unfortunately the algorithm need not terminate with a singleton set (assuming that S contains the solution). Consider $e = \mu x.(x \sqcup \nu y.(x \sqcap y))$ with corresponding equations

$$\mu x = \max\{x, y\},$$
$$\nu y = \min\{x, y\}.$$

The set of all candidate solutions is $S = \{(0,0),(1,1)\}$. Then $S_2 = \{(0,0),(1,1)\}$ and $S_1 = S_2$. Nevertheless, in many cases the process does indeed lead to a singleton set or at least a smaller set of canditates. In pragmatic terms it is wroth running before using the recursive algorithm of §6.

The selection step of the algorithm involves solving a single variable fixed point problem. This can be carried out using the algorithm of §5 or by the process given in the next section, the worst case cost is bounded by a single exponential in the size of the equation. The method presented in the next section has the potential to terminate quickly.

7. Terms in one variable

As is well known the solution to $\mu x.f(x)$ can be obtained as the limit of the iteration $f(0), f^2(0), f^3(0), \ldots$ (and similarly for $\nu x.f(x)$ by replacing the argument 0 with 1), see Arnold and Niwiński [2]. Unfortunately this might not converge after finitely many steps even for simple expressions. Consider, for example, $\mu x.(1/2 \sqcup (2/3\, x \oplus 1/3))$. Setting $f(x) = 1/2 \sqcup (2/3\, x \oplus 1/3)$ we see easily that

$$f(x) = \begin{cases} \frac{1}{2}, & \text{for } x \in [0, \frac{1}{4}]; \\ \frac{2}{3}x + \frac{1}{3}, & \text{for } x \in [\frac{1}{4}, 1]. \end{cases}$$

Clearly $f(x)$ has exactly one fixed point at $x = 1$. Now

$$f^{n+1}(0) = \frac{1}{2}\left(\frac{2}{3}\right)^n + \frac{1}{3}\left(\left(\frac{2}{3}\right)^{n-1} + \left(\frac{2}{3}\right)^{n-2} + \cdots + 1\right) = 1 - \frac{1}{2}\left(\frac{2}{3}\right)^n,$$

for all $n \geq 0$. Thus the iteration converges to 1, as stated by the Knaster–Tarski theorem, but not after finitely many steps. The same will happen if the iteration reaches any line segment with gradient strictly less than 1.

Consider a term $\sigma x.e$ where e involves only x as a bound variable and has no free variables. The term e denotes a piecewise linear function so that if we express it explicitly as such then it is easy to find the least or greatest fixed point. Unfortunately the number of pieces can be exponential in the size of e. Thus we not only face exponential time by this approach but also exponential space. If we use the translation of §4 we obtain a polynomial $f(x, t_1, \ldots, t_m)$ that is affine linear in x and affine multilinear in $t_1, \ldots, t_m$. We also obtain side conditions $\mathrm{leq}(g_1, t_1)$,..., $\mathrm{leq}(g_m, t_m)$ where g_i involves only $t_1, \ldots, t_{i-1}$ and x. The expressions are of the same order of size as e. We can find the relevant fixed point by trying all 2^m assignments of $t_1, \ldots, t_m$ but this commits us to exponential time no matter what happens.

It is possible to combine the two approaches and often avoid exponential time as follows. For the sake of definiteness assume that we want the least fixed point. First of all assume $x = 0$. This determines the value of t_1. Substituting this into the remaining inequalities we determine the values of $t_2, \ldots, t_m$ in turn. This yields m inequalities involving only x and rationals; the inequalities are consistent due to the choice of the values of the t-variables. We thus obtain a closed interval $[0, a_1]$ over which these inequalities do not change. Substituting the values of $t_1, \ldots, t_m$ into $f(x, t_1, \ldots, t_m)$ yields an affine linear function in x alone thus we can determine if it has a fixed point and if so determine the least such. If there is such a point then we are done. Otherwise we consider x to have the value $a_1 + \epsilon$ where ϵ is an arbitrarily small positive number (an infinitesimal). This now yields new values for the t-variables and a closed interval $[a_1, a_2]$ over which the inequalities do not change. Once again we can determine if there is a fixed point. Continuing in this way we are guaranteed to find the least fixed point. Moreover we need only ever keep the end point of the last interval constructed in order to construct the next one. This method is efficient provided the fixed point occurs on an early linear piece of the piecewise linear function denoted by the term.'

Example. Consider the term $e = \mu x.((1/2\, x \sqcup 1/4) \sqcup (x \sqcap 3/8))$. The body of this translates to

$$f = (1 - t_3)(1/2\,(1 - t_1)x + 1/4\, t_1) + t_3(t_2 x + 3/8 - 3/8\, t_2),$$

with side conditions

$$\mathrm{leq}(1/2\, x, 1/4, t_1),\ \mathrm{leq}(x, 3/8, t_2),\ \mathrm{leq}(1/2\,(1-t_1)x+1/4\, t_1, t_2 x+3/8-3/8\, t_2, t_3).$$

Setting $x = 0$ we obtain $t_1 = 1$ and $t_2 = 1$ from the first two side conditions. The third condition now becomes $\mathrm{leq}(1/4, x, t_3)$ and so $t_3 = 0$. To sum up

$$t_1 = 1,\ t_2 = 1,\ t_3 = 0;$$

$$f = 1/4;$$

$$x \leq 1/4,\ x \leq 3/8,\ x/2 \leq 1/4.$$

Thus the interval over which the inequalities do not change is $[0, 1/4]$. this yields $x = 1/4$ as the first, and hence least, fixed point.

As an illustration we construct the whole piecewise linear function. Setting $x = 1/4 + \epsilon$ we obtain again that $t_1 = 1$, $t_2 = 1$ but this time $t_3 = 1$. The situation is now

$$t_1 = 1,\ t_2 = 1,\ t_3 = 1;$$

$$f = x;$$

$$1/4 \leq x,\ x \leq 3/8,\ x/2 \leq 1/4.$$

The new interval is $[1/4, 3/8]$.

For the next stage we set $x = 3/8 + \epsilon$, obtaining $t_1 = 1$, $t_2 = 0$ and $t_3 = 1$. The situation is now

$$t_1 = 1,\ t_2 = 0,\ t_3 = 1;$$

$$f = 3/8;$$

$$1/4 \leq 3/8,\ 3/8 \leq x,\ x/2 \leq 1/4.$$

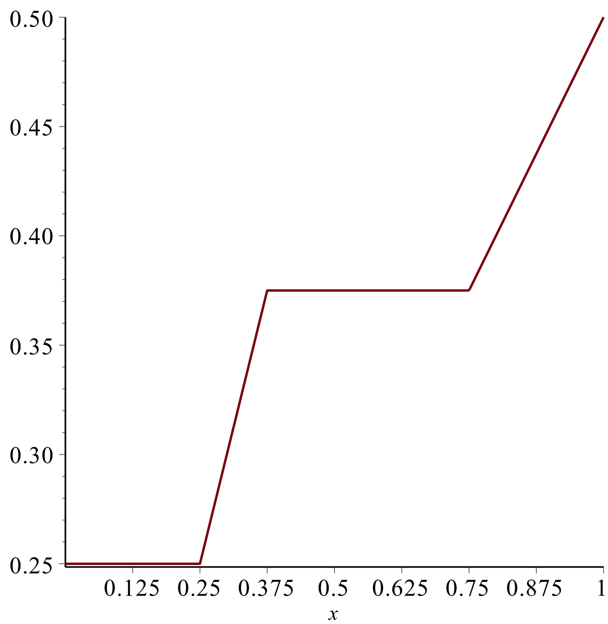

Figure 2: Plot for the function of $(1/2\,x \sqcup 1/4) \sqcup (x \sqcap 3/8)$.

The new interval is $[3/8, 1/2]$. There is thus another fixed point at $x = 3/8$.

For the next stage we set $x = 1/2 + \epsilon$, obtaining $t_1 = 0$, $t_2 = 0$ and $t_3 = 1$. The situation is now

$$t_1 = 0,\ t_2 = 0,\ t_3 = 1;$$

$$f = 3/8;$$

$$x/2 \leq 3/8,\ 3/8 \leq x,\ 1/4 \leq x/2.$$

The new interval is $[1/2, 3/4]$.

Now we set $x = 3/4 + \epsilon$, obtaining $t_1 = 0$, $t_2 = 0$ and $t_3 = 0$. The situation is now

$$t_1 = 0,\ t_2 = 0,\ t_3 = 0;$$

$$f = x/2;$$

$$3/8 \leq x/2,\ 3/8 \leq x,\ 1/4 \leq x/2.$$

The new interval is $[3/4, 1]$. The graph of the function is shown in Figure 2.

By completing the process we have also shown that the greatest fixed point is at $x = 3/8$, however this is not a good way to find it. For this we would start at $x = 1$ and if the current interval is $[a, b]$ we set $x = a - \epsilon$ to find the next one.

Finally, note that the process described does not necessarily yield the smallest number of pieces; in the example the third and fourth ones can be combined into one.

Acknowledgment: The author is grateful to Alex Simpson for suggesting the problem as well as Matteo Mio for early comments. He is also grateful to the two anonymous referees whose helpful comments have improved the paper.

References

[1] M. Mio, A. Simpson, Łukasiewicz μ-calculus, Fundamenta Informaticae 150 (3-4) (2017) 317–346.

[2] A. Arnold, D. Niwiǹski, Rudiments of μ-calculus, Elsevier, 2001.

[3] J. Ferrante, C. Rackoff, A decision procedure for the first order theory of real addition with order, SIAM Journal of Computing 4 (1) (1975) 69–76.

[4] B. Boigelot, S. Jodogne, P. Wolper, An effective decision procedure for linear arithmetic over the integers and reals, ACM Transactions on Computational Logic 6 (13) (2005) 614–633.

[5] D. Kozen, Results on the propositional μ-calculus, Theoretical Computer Science 27 (1983) 333–354.

[6] K. Kalorkoti, Model checking in the modal μ-calculus and generic solutions, J. Symbolic Computation 46 (2011) 584–594.

[7] J. von zur Gathen, J. Gerhard, Modern Computer Algebra, 2nd Edition, Cambridge University Press, 2003.

[8] E. H. Bareiss, Sylvester's identity and multistep integer-preserving gaussian elimination, Mathematics of Computation 22 (1968) 565–578.